
pyramid-jinja2-webpack

Release 0.1.1

Jun 03, 2017

Contents

1 Overview	1
1.1 Installation	1
1.2 Documentation	1
1.3 Development	1
2 Installation	3
3 Usage	5
4 Configuring jinja2_webpack	7
5 Referencing assets from template	9
6 Referencing assets from python	11
7 More Info	13
8 Reference	15
8.1 pyramid_jinja2_webpack	15
9 Contributing	17
9.1 Bug reports	17
9.2 Documentation improvements	17
9.3 Feature requests and feedback	17
9.4 Development	18
10 Authors	19
11 Changelog	21
11.1 0.1.0 (2017-06-01)	21
12 Indices and tables	23
Python Module Index	25

CHAPTER 1

Overview

docs	
tests	
package	

Pyramid integration of jinja2_webpack

- Free software: BSD license

Installation

```
pip install pyramid-jinja2-webpack
```

Documentation

<https://pyramid-jinja2-webpack.readthedocs.io/>

Development

To run the all tests run:

```
tox
```

To run on a specific version of python run e.g.:

```
tox -e py34  
tox -e py27
```

CHAPTER 2

Installation

At the command line:

```
pip install pyramid-jinja2-webpack
```


Include the project into pyramid:

```
pyramid.includes=  
    ...  
    pyramid_jinja2_webpack
```

Or:

```
def main(global_config, **settings):  
    config = Configurator(settings=settings)  
    config.include('pyramid_jinja2_webpack')
```


CHAPTER 4

Configuring jinja2_webpack

You can put settings into the ini:

```
webpack.manifest=/path/to/manifest.json  
webpack.publicRoot=/static/pack/  
webpack.errorOnInvalidReference=false
```

Referencing assets from template

In your jinja2 assets you can refer to webpack entries by URL using the filter syntax:

```
{{ 'my-entry': webpack }}
```

Referencing assets from python

You can access the webpack environment from either the config object:

```
webpack_env = config.get_webpack_environment()
```

Or from the request object:

```
webpack_env = request.webpack_environment
```

You can then query an asset like:

```
asset = webpack_env.identify_asset_spec('my-entry')  
url = asset.url
```


CHAPTER 7

[More Info](#)

See: [jinja2_webpack](#).

pyramid_jinja2_webpack

exception `pyramid_jinja2_webpack.Jinja2EnvironmentMissingException`

Thrown when configuration fails because it can't find the jinja2 environment

Contributions are welcome, and they are greatly appreciated! Every little bit helps, and credit will always be given.

Bug reports

When [reporting a bug](#) please include:

- Your operating system name and version.
- Any details about your local setup that might be helpful in troubleshooting.
- Detailed steps to reproduce the bug.

Documentation improvements

pyramid-jinja2-webpack could always use more documentation, whether as part of the official pyramid-jinja2-webpack docs, in docstrings, or even on the web in blog posts, articles, and such.

Feature requests and feedback

The best way to send feedback is to file an issue at <https://github.com/JDeuce/python-pyramid-jinja2-webpack/issues>.

If you are proposing a feature:

- Explain in detail how it would work.
- Keep the scope as narrow as possible, to make it easier to implement.
- Remember that this is a volunteer-driven project, and that code contributions are welcome :)

Development

To set up *python-pyramid-jinja2-webpack* for local development:

1. Fork *python-pyramid-jinja2-webpack* (look for the “Fork” button).
2. Clone your fork locally:

```
git clone git@github.com:your_name_here/python-pyramid-jinja2-webpack.git
```

3. Create a branch for local development:

```
git checkout -b name-of-your-bugfix-or-feature
```

Now you can make your changes locally.

4. When you’re done making changes, run all the checks, doc builder and spell checker with *tox* one command:

```
tox
```

5. Commit your changes and push your branch to GitHub:

```
git add .
git commit -m "Your detailed description of your changes."
git push origin name-of-your-bugfix-or-feature
```

6. Submit a pull request through the GitHub website.

Pull Request Guidelines

If you need some code review or feedback while you’re developing the code just make the pull request.

For merging, you should:

1. Include passing tests (run *tox*)¹.
2. Update documentation when there’s new API, functionality etc.
3. Add a note to *CHANGELOG.rst* about the changes.
4. Add yourself to *AUTHORS.rst*.

Tips

To run a subset of tests:

```
tox -e envname -- pytest -k test_myfeature
```

¹ If you don’t have all the necessary python versions available locally you can rely on Travis - it will run the tests for each change you add in the pull request.

It will be slower though ...

CHAPTER 10

Authors

- Josh Jaques - <https://jdeuce.net>

CHAPTER 11

Changelog

0.1.0 (2017-06-01)

- First release on PyPI.

CHAPTER 12

Indices and tables

- `genindex`
- `modindex`
- `search`

p

pyramid_jinja2_webpack, 15

J

Jinja2EnvironmentMissingException, 15

P

pyramid_jinja2_webpack (module), 15